Automated Planning for Spacecraft and Mission Design

Ben Smith
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
firstname.lastname@jpl.nasa.gov

Barbara Engelhardt
Jet Propulsion Laboratory

Russell Knight
Jet Propulsion Laboratory

Darren Mutz
Jet Propulsion Laboratory

ABSTRACT

Mission design engineers identify a spacecraft design and mission plan that best achieves the mission objectives while staying within cost, mass, and operability constraints. It is often easiest to evaluate a spacecraft design in the context of a detailed mission plan. Generating plans by hand is labor-intensive. We present an AI planning system that automatically generates and evaluates mission plans for specified spacecraft designs. This system has been applied to design problems from a number of NASA missions.

KEYWORDS: planning, scheduling, AI, design optimization, mission planning.

INTRODUCTION

The job of mission design engineers is to identify a spacecraft design and mission plan that best achieves the mission objectives while staying within cost, mass, and operability constraints. We observe that it is often easier to evaluate a spacecraft and mission design in the context of activity plans for key mission scenarios. Just as a simulation allows designers to better understand how the design artifact would behave, a plan helps mission designers to understand how a specified spacecraft design will execute a given mission scenario. For example: How many observations will it take? What are the resource margins? How much slack time is there for contingencies?

We have developed an automated planning system that takes as input spacecraft parameters (e.g., spacecraft slew rates, battery capacity) and mission parameters (e.g., observation requests, frequency of communication passes, trajectory). The planner generates a mission activity plan that achieves the mission goals while obeying the

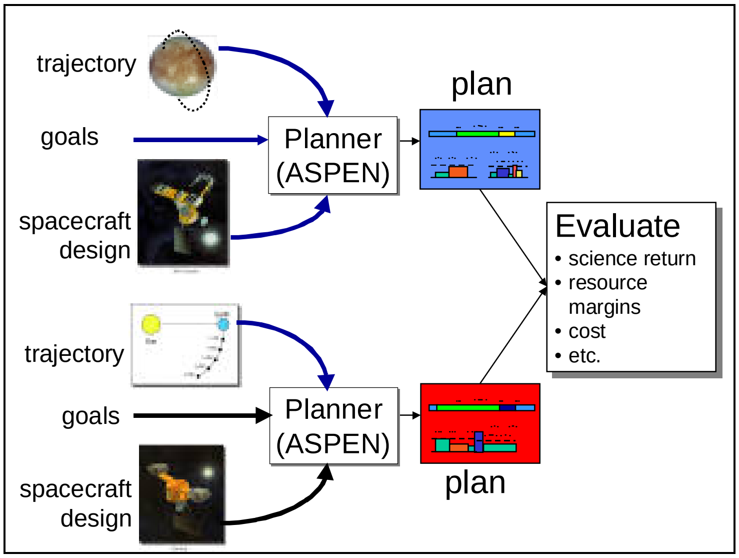

Figure 1. Architecture

constraints imposed by the given mission and spacecraft design (which are a function of the mission and spacecraft parameters).

This technology enables mission engineers to quickly evaluate several designs. Engineers can evaluate several candidate designs against a given mission scenario by generating plans for each design and automatically evaluating them against objective criteria. Engineers can also use this system for "what-if" evaluations. They can see how a given designs performs in the context of a mission scenario, and then modify the design or mission to improve performance. For example, a spacecraft may be limited to ten science images per orbit because of insufficient on-board data storage, even though there are opportunities for several. The engineer increases the memory parameter and generates a new plan to see if the spacecraft can now take more science images.

This system has performed design evaluations for several NASA missions. The remainder of this paper will describe the system in more detail, provide some example trade studies, and discuss the key scheduling issues and algorithms.

SYSTEM ARCHITECTURE

The system takes as input spacecraft and mission design parameters and a set of scenario goals. From these inputs an automated planner produces a plan of spacecraft activities that accomplish the scenario goals in a way that is consistent with the spacecraft and mission design. The next subsection describes this process in more detail. The resulting plan is then evaluated with respect to user-specified objective criteria. The overall architecture is shown in Figure 1.

Automated Planning

The core of this system is an automated planning and scheduling system. We used the Aspen [1] planner, which has a number of reasoning capabilities we find necessary for generating spacecraft mission plans. However, the architecture makes no assumptions about the planner, so one could easily substitute a different planning system.

```
Activity TakeImage {
    int target-id;
    constraints=
            starts_after end_of turn-instr-on by [30, infinity];
    resources =
            storage use 20, // Mb
            power use 10,  // Watts
};

Resource storage { type=depletable; capacity=100; // Mb }
Resource power   { type=non-depletable; capacity = 130; // Watts }
```

Figure 2. Model Fragment

An automated planner, such as Aspen, takes as input a set of goals and an initial state. It then derives a set of actions that will achieve the goals from the initial state. A *domain model* specifies the available actions, states, resources and constraints among them. For example, a “take-science-image” action may require that the spacecraft be in the low-vibration state, occur at least 30s after turning on the instrument, and requires 20Mb of on-board storage and 10W of power. These constraints are specified in a declarative language specific to the planning system. The above constraints would be specified in Aspen as shown in Figure 2.

Aspen generates a plan that achieves the goals from the initial state while obeying the constraints in the domain model. It does this by a combination of sub-goaling, goal-expansion, and conflict resolution. Sub-goaling achieves a desired state by identifying an activity that achieves the state and inserting it into the plan. Goal-expansion takes a high-level goal and expands it into a pre-defined set of sub-goals. Conflict resolution identifies constraint violations in the plan and resolves them. For example, the plan may contain more “take-science-image” activities than will fit onto the on-board storage. Aspen might resolve this conflict by removing some of the images or inserting a “downlink” activity that will free up more onboard storage. Aspen uses an algorithm called *iterative repair* [4] to perform conflict resolution.

Design Parameters

The domain model specifies the constraints for a given spacecraft design. The goals specify the mission scenario. Each spacecraft design needs a different model, and each mission scenario needs a different set of goals. However, the spacecraft and mission designs are often similar in many ways. To avoid generating new models and goals for each design, we parameterize them with appropriate design variables. The user simply specifies values for each of the design variables, which results in an appropriate model and set of goals. This allows the designer to explore the parameter space quickly and easily. It also opens the possibility of searching that space automatically for an optimal design. This is an area for future research.

Some typical spacecraft design parameters are resource capacities (battery, on-board data storage, fuel), operability constraints (how long does it take to warm-up the instrument, how much data storage does an image require), and hardware options (use cold-gas thrusters or a reaction wheel). These variables can all be expressed as parameters in the domain model.

```
Activity TakeImage {
    int target-id;
    int PowerAmount, StorageAmount, InstrOnDelay;
    constraints=
            starts_after end_of turn-instr-on by [InstrOnDelay, infinity];
    resources =
            storage use StorageAmount, // Mb
            power use  PowerAmount,  // Watts
};
```

Figure 3. Parameterized Activity

The Aspen domain language allows parameters in constraints and activities. An example of a parameterized activity is shown in Figure 3. The parameters are in *italics.*

The user specifies values for the design variables, and the parameters in the model are set accordingly. This can be done a number of ways: a global search-and-replace on the model file; adding equivalence constraints between the activity parameters and the parameters of a dummy goal activity that specifies the values; and declaring that the parameter values are determined by an external function that returns a value specified in a file of parameter values. We have primarily used the third method for Aspen, though other methods may be more appropriate for other planning systems (especially ones without external dependency functions) or for performance reasons.

The mission scenario is specified by a set of goals. Goals are simply a set of activities that must appear in the plan. The user specifies one set of goals for each mission scenario. The goals have parameters that can be design variables.

Plan Evaluation

Plans are evaluated with respect to user-specified evaluation criteria. A new evaluation function must be written for each criteria. Some typical evaluation criteria are resource margins, resource usage, and science return.

APPLICATIONS

Space Interferometry Mission

The planning for mission design system supported an orbit trade study for the Space Interferometry Mission (SIM). The question was whether to use an inexpensive but highly constraining low-Earth orbit, or a more expensive but less constraining Earth-trailing orbit.

SIM will use a space-borne interferometer to take images of distant stars with much higher resolution than is possible with existing telescopes. One of the key scenarios in this mission is a “grid campaign” where the spacecraft images the entire celestial sphere over a period of about a month. To minimize the length of the grid campaign, and thereby maximize the science return, the images must be ordered to minimize the angular distance between adjacent targets. To avoid damage, the interferometer must not be pointed within a certain angular distance (the *exclusion angle*) of bright bodies in the solar system, such as the Sun, Earth, moon, Mars and Jupiter. Over time targets move in and out of exclusion angles relative to the spacecraft as determined by the spacecraft orbit and celestial mechanics. For a fixed trajectory, each target can therefore be imaged only during time windows when it is not in the exclusion angle. In general the more exclusion windows there are the longer the optimal tour becomes, but it is difficult to say how much longer without actually solving them.

We used the Aspen planner to generate a grid campaign for the Earth-trailing and low-Earth orbit cases, and for different exclusion angles. The objective was to determine whether Earth-trailing campaigns, which have fewer exclusion windows than Earth-orbit campaigns, were sufficiently faster to justify the more expensive orbit. The results, shown in Table 1, supported the decision to use an Earth-trailing orbit. Targets is the total number of image targets in the campaign, scheduled is the number of targets that could be taken (some targets are never visible, or their widows overlap so that there is only enough time to take some of them). The plan duration is the total duration of the grid campaign as planned, and time/target is the plan duration divided by the number of scheduled targets.

Table I. SIM Plan Evaluation Summary.

Trajectory	targets	scheduled	plan duration	time/target
Earth-orbit	1164	1141	25.23 days	31.84 min
Earth-trail	1164	987	26.78 days	39.07 min

LightSAR

The LightSAR mission is an Earth-orbiting satellite with a synthetic aperture radar (SAR). The SAR footprint is a rectangular swath over the Earth's surface. The objective is to image specified regions of the Earth (say Greenland) within certain time windows (e.g., March to June). To image a region, one must select a set of rectangular swaths that cover the region. The available swaths and the time at which each swath can be taken depend on the spacecraft orbit and the SAR beam angle (there are several adjacent beams with incidence angles separated by a few degrees). For example, one might be able to image a given strip of Greenland from 7:00 am to 7:05 am on Beam 5, or from 7:15 am to 7:23am on Beam 3. Each swath results in many megabytes of data, which reside on the on-board recorder until it can be downlinked. The planning problem is to select beams that cover the desired regions within the specified time windows without exceeding the on-board storage.

The design questions are how the on-board storage constraints and downlink opportunities impact the science return. The storage capacity and downlink opportunities limit the number of swaths per orbit, and thus the total science return, but in a manner that is hard to predict. By generating plans for various storage capacities, available downlink stations, and goal distributions we can understand that relationship and pick values that provide the best balance between science return and cost.

These plans take weeks to generate by hand, but only minutes with an automated planner. Planning technology makes it feasible to explore relationships like this and thereby improve the design in ways that would otherwise not be possible. We have generated plans for the baseline design, and are beginning to explore this relationship in more detail.

Planning Challenges

The planning problems described above are uniquely challenging. The overall problem has many constraints that require a powerful planning system like Aspen. For example, SIM must also consider battery and power constraints and the need to periodically "decondition" the reaction wheel (bleed off excess momentum); LightSAR has interferometry pairs, which are pairs of SAR images that must be take exactly 10 days apart, and a number of miscellaneous constraints on the instrument, data recorder, and downlink activities. The core problem is often a combinatorial optimization problem. For example, the core SIM problem is an instance of the traveling salesman problem with time windows [3], and the LightSAR problem is a kind of constrained bin-packing problem [2]. These often require specialized algorithms to solve effectively (e.g., [2,3]). General-purpose planners often solve them poorly or slowly because they do not have these specialized data structures or algorithms.

We addressed this problem by mapping an *abstraction* of the planning problem to the core combinatorial optimization problem, and solving that core problem with a special-purpose solver. The additional constraints are expressed in the core problem as a general feasibility constraint into which the specialized solver has no visibility. The solution to this problem then guides the planner in solving the overall planning problem. In our experience this approach yields high-quality solutions within reasonable computational bounds (a few minutes to a few hours, depending on the problem).

CONCLUSION

It is often easier to evaluate a spacecraft design in the context of a mission scenario. We have developed a planning system that automatically generates mission plans for specified designs. The system can generate scenario plans in minutes that would take designers weeks to generate by hand. This allows designers to more quickly explore the design space and to see interactions between spacecraft design and operations that would be difficult to identify by other means.

This system has been applied to design problems for a number of spacecraft missions and has met with enthusiasm from the design engineers.

ACKNOWLEDGEMENTS

This paper describes work performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract from the National Aeronautics and Space Administration.

REFERENCES

1. Fukunaga, A.; Rabideau, G.; Chien, S.; and Yan, D. "Toward an Application Framework for Automated Planning and Scheduling," in *Proceedings of the 1997 International Symposium on Artificial Intelligence, Robotics, and Automation for Space*, Tokyo, Japan (1997).
2. Garey and Johnson. *Computers and Intractability: A guide to the theory of NP-completeness.* W.H. Freeman and co., New York (1979)
3. Solomon, M. "Algorithms for the vehicle routing and scheduling problems with time window constraints." *Operations Research* **35**(2) (March-April, 1987), 254-265.
4. Zweben, M.; Daun, B.; Davis, E.; and Deale M., "Scheduling and Rescheduling with Iterative Repair," in Intelligent Scheduling, Morgan Kaufman, San Francisco (1994).